Software Requirements Specification

Version 1.0

October 30, 2024

TradeNest

Kaman Wong
Jason Opoku
Naomi Teklu
Shalom Aideyan

Table of Contents

1. Introduction

1.1 Purpose

This document sets out the overarching software requirements for the TradeNest system, a platform designed to streamline the exchange, lending, selling and recovery of lost and found items among university students. The purpose of this document is to explicitly define the core functionality, intended users and capabilities of the system from a requirements perspective. It describes what the system will achieve rather than how it will be implemented. This document is intended for project stakeholders, including project developers, designers, testers, and academic advisors, to provide them with a clear understanding of the system's goals and functionality.

1.2 Scope

1.2.1 In Scope

1. User-to-User Communication:

Users can communicate directly through in-app messaging.

2. Lost Item Posting:

Users who find lost items can post them with pictures and descriptions.

3. Institutional Email Registration:

Account creation is limited to users with an institution email (@georgebrown.ca).

4. Item Management:

Users can view, edit, and manage their posted items.

5. Search and Filter:

Users can search for items posted by other users and apply filters to refine results based on preferences.

6. Transaction History and Feedback:

Users can view past transactions and provide feedback on them.

7. Profile Customization:

Users can personalize their profiles by adding a picture, description, and contact information.

8. **Favoriting and Following:**

Users can favorite items and follow other users for easier access to desired items.

1.2.2 Out of Scope

1. Lost Item Reporting by Owners:

Users who have lost items cannot post them; only found items can be reported.

2. Personal Email Registration:

Account creation is restricted to institutional email addresses and does not support personal emails.

3. Online Payment Integration:

The platform does not support payments for traded items.

4. Online Payment Integration:

The system is institution-specific and will not work across multiple institutions.

5. Biometric Authentication:

The system will not support fingerprint, facial recognition, or other biometric login methods.

6. Voice or Video Calling:

Users can only communicate through text messaging; calling features are excluded.

7. Delivery Services:

The system will not facilitate the delivery of traded items; users are responsible for arranging item exchanges.

8. Multi-Language Support:

The platform will be available only in English to cater to the primary user base.

1.3 References

Reference File Name	Version
F24_T47_Project Vision	1.0
F24_T47_ PersonasAndUserStories	1.0
F24_T47_Project_Plan	1.0

2. System Overview

2.1 Project Perspective

TradeNest app is a new self-contained system that operates independently to provide users a platform to conduct various tasks within their social environment. Certain features would be implemented to allow users to perform their desired outcomes regarding selling, trading, lending as well as a lost-and-found system.

2.2 System Context

The TradeNest app addresses several issues that'll benefit all parties involved. For instance, the chances of a lost item being returned to its rightful owner would increase due to the aid of a platform such as TradeNest, allowing users to communicate among one another. Our platform also limits the number of apps needed to conduct multiple actions as our singular application handles multiple purposes like lending items, trading and selling. Wasted products continuously being brought as replacements would decrease which would increase morale among users.

2.3 General Constraints

- TradeNest would be constrained to comply with data regulations necessary for it to function smoothly
- TradeNest must include an interface that aligns with the branding of the institution
- TradeNest is institution-specific, thus catering to that specific institution alone

2.4 Assumptions and Dependencies

2.4.1 Assumptions

- Users must have access to compatible devices and internet to interact with the app
- Users must be willing to engage and connect with others
- Users must have an institution email address to gain access to their accounts

2.4.2 Dependencies

- Compliance with the institution would be necessary for the application to function as purposed

3. Functional Requirements

3.1 Feature Requirements

3.1.1 User Registration

Feature	User Registration
Introduction	Users can register and create an account using the school's email to ensure that the platform is only accessible to students, staff and faculty within the school.
Inputs	- School email address - Password
Processing	The system verifies that the incoming email belongs to the school domain. The system creates the account and stores it in the database. The system sends a verification email to the user's email for further confirmation.
Outputs	- Successful registration confirmation message - Verification email notification.

3.1.2 Item Posting of Sale, Barter and Rental

Feature	Item Posting of Sale, Barter and Rental
Introduction	Users can post items to be traded, providing names, descriptions, pictures and categories for other students to view.
Inputs	- Item name - Description - Price - Classification - Pictures
Processing	The system checks the integrity of the input data. The system saves the item data to the database and publishes it on the platform.
Outputs	- Successfully publish a notification - Display the item on the platform

3.1.3 Item Search and Filter

Feature	Item Search and Filter
Introduction	Users can use filters to search for items from other users.
Inputs	- Keywords - Categories - Price
Processing	The system filters items in the database based on the user's search criteria. The system generates a list of items that match the criteria.
Outputs	- List of eligible items

3.1.4 Lost Item Posting

Feature	Lost Item Posting
Introduction	Users can post "Found Items" to help the owner to retrieve the lost items.
Inputs	- Photos - Location - Description
Processing	The system checks for data integrity. The system stores the information of the found items in the database and publishes it in the "Found" list.
Outputs	- Success notification - Display the item in the Found list

3.1.5 Messaging System

Feature	Messaging System
Introduction	Users can contact other users directly through the messaging system on the platform to conduct transactions or exchanges.
Inputs	- Select the user - Message content
Processing	The system saves the message and sends it to the recipient. The recipient is notified and can view the message.
Outputs	- Message notification

3.1.6 User Profile Customization

Feature	User Profile Customization
Introduction	Users can edit their personal information to personalise their accounts, including photos, descriptions and contact details.
Inputs	- Photos - Descriptions - Contact information
Processing	The system validates and stores the updated data.

	The User's profile is updated and visible on the platform.
Outputs	- Notification of successful update - Display of new information

3.1.7 Transaction History and Feedback

Feature	Transaction History and Feedback
Introduction	Users can view past transactions and provide ratings for transactions to build trust in the platform.
Inputs	- Selects the transaction history - Enters a rating.
Processing	The system stores the evaluation in the transaction record. The rating is displayed in the transaction log and affects the user's score.
Outputs	- Notification of successful appraisal submission - Update of transaction history

3.1.8 Favorites and Following

Feature	Favorites and Following
Introduction	Users can mark favorite items as favorites or track other users for quick access.
Inputs	- Selected items for collection - Selected user for tracking
Processing	The system stores user information about collected items or tracked users in the user's account. Collected items and tracked users are displayed on the user's personal page.
Outputs	- Bookmarking and tracking success notifications - Updating the user's favorites list.

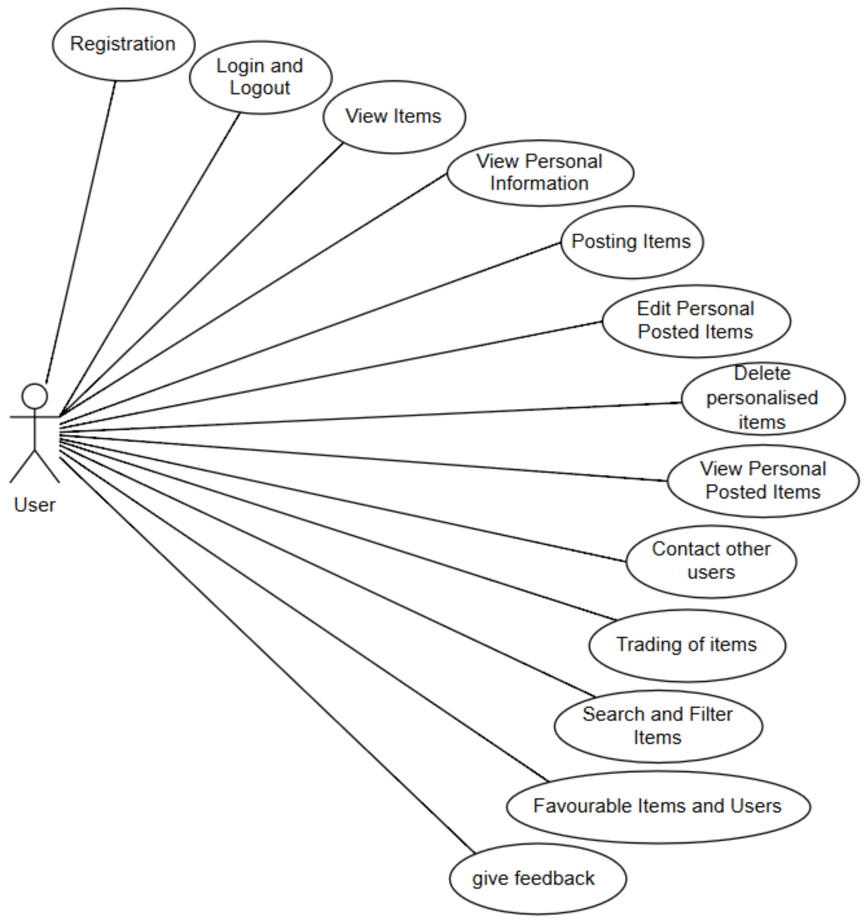
3.1.9 Notifications System

Feature	Notifications
Introduction	Users are notified when transactions are completed, messages are received, etc. for immediate response.
Inputs	- System triggered notification events (e.g. message or transaction completion)
Processing	The system generates notification messages and sends them to the relevant users. Users can view the notification content immediately.
Outputs	- Receive notification messages - View them on the platform.

3.2 Use Cases

3.2.1 User Use Cases

Diagram:



Use Case: Registration

Brief Description:

Users can sign up for an account and must use the school's email for authentication.

Initial Step-By-Step Description:

Prior to the activation of this use case, the user has visited the registration page of the TradeNest platform.

1. User selects ‘Register’.

2. The system displays the registration form.

3. User enters school email, password.

4. User submits the form.

5. The system verifies that the email is valid and creates an account, then sends a verification email.

Alternative Paths:

If the school email entered is invalid, the system indicates the error and requests a new entry.

If the email is already registered, the system prompts the user that the account already exists and suggests login.

Use Case: Login and Logout

Brief Description:

Users can log in and out of the Platform to protect their personal information.

Initial Step-By-Step Description:

Before this use case is launched, the user visits the TradeNest login page.

1. User enters school email and password.

2. The system verifies the account information.

3. If the account is correct, the system allows the user to log in and displays the home page.

4. The user completes the session and selects Logout.

5. The system ends the user's login session.

Alternative Paths:

If the account or password entered is incorrect, an error message will be displayed, and you will be asked to re-enter it.

Use Case: View Items

Brief Description:

Users can browse the list of items posted by other users and view the details of the items.

Initial Step-By-Step Description

When the use case is activated, the user is logged into the system and accesses the Browse Items page.

1. the user selects Browse Items (e.g. Sale, Barter, Rental and Lost).

2. The system displays a list of all items.

3. The user selects a specific item to view the details.

4. The system displays a detailed description of the item, its price, or the terms of the transaction.

Use Case: View Personal Information

Brief Description:

Users can view their personal information, including account information and settings.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and goes to the ‘Profile’ page.

1. The user selects Personal Information.

2. The system displays the account's personal information and settings.

3. The user can check and view the account details.

Alternative Paths:

If personal information is incomplete, the user is prompted to add or update the information.

Use Case: Posting Items

Brief Description:

Users can post information about trades or find items for other users to view.

Initial Step-By-Step Description:

Before the use case is activated, the user is logged in and goes to the ‘Release Items’ page.

1. The user selects Release Items.

2. The system displays the posting form.

3. The user enters the item name, description, price, and category, and uploads a photo.

4. The user submits the form.

5. The system saves and displays the item in the item list.

Alternative Paths:

If you do not fill in the required information (e.g., description or classification) for an item, the system will indicate an error and ask for additional information.

If the upload fails, the system will indicate an error and allow the user to re-upload.

Use Case: Edit Personal Posted Items

Brief Description:

Users can edit published item information to update descriptions or conditions.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and goes to the ‘My Items’ page.

1. The user selects an item to edit.

2. The system displays the edit form for the item.

3. The user updates the item information.

4. the user submits the changes.

5. the system saves and updates the item information.

Alternative Paths:

If the updated information lacks the necessary content, the user is prompted to add the information.

Use Case: Delete Personalized Items

Brief Description:

Users can delete published items to remove them from the Platform.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and goes to the ‘My Items’ page.

1. The user selects the item to be deleted.

2. The system asks for confirmation of the deletion.

3. The user confirms the deletion.

4. The system deletes the item and removes it from the platform.

Alternative Paths:

If the user cancels the delete operation, the system keeps the item displayed on the platform.

Use Case: View Personal Posted Items

Brief Description:

Users can view published items for easy management and editing.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and goes to the ‘My Items’ page.

1. The user selects My Items.

2. The system displays all the items that the user has posted.

3. The user can choose to view, edit, or delete the item.

Alternative Paths:

If the user does not have a published item, the system displays the message "No published item".

Use Case: Contact Other Users

Brief Description:

Users can connect with other users through the messaging system on the platform.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and viewing the item details page of other users.

1. The user selects Contact Seller or Contact Renter.

2. The system displays a message input box.

3. The user enters the message content and clicks Send.

4. The system sends the message to the target user.

Alternative Paths:

If the message fails to be sent, the system indicates an error and allows the user to retry.

Use Case: Trading of Items

Brief Description:

Users can complete transactions with other users and mark the transaction status as completed.

Initial Step-By-Step Description:

When the use case is activated, both parties have contacted each other and agreed to carry out the transaction.

1. Both parties confirm the transaction

2. After the transaction is completed, the system updates the transaction status to ‘Completed’.

4. the user can view the transaction in the transaction history.

Use Case: Favorable Items and Users

Brief Description:

Users can mark items or other users as favorites for quick access.

Initial Step-By-Step Description:

When the use case is activated, the user is logged in and viewing a list of items or users.

1. The user selects ‘Favorite Items’ or ‘Track Users’.

2. The system adds the selected item or user to the favorites list.

3. The system displays a notification that the favorite is successful.

4. the user can view the favorite items and users from his/her personal favorites.

Alternative Paths:

If the user has bookmarked the item or has tracked the user, the system displays an alert message.

Use Case: Search and Filter Items

Brief Description:

Users can search for items using keywords, categories, and filters to help quickly find items that match their needs.

Initial Step-By-Step Description:

When the use case is activated, the user is logged into the system and accesses the Search Items page.

1. The user enters the Search Items page.

2. The system displays search criteria options, including a keyword input box, category selection, and filters (such as price range or release date).

3. The user enters keywords, selects a category, or sets a filter.

4. The user clicks the ‘Search’ button.

5. The system filters the items in the database according to the user's search criteria and generates a list of eligible items.

6. The system displays a list of eligible items for the user to view.

Alternative Paths:

If no matching items are found, the system displays a "no results found" message and suggests the user to adjust the search criteria.

Postcondition:

Users can browse through the list of items matching the search criteria to view further details of the item or contact the owner.

Use Case: Give Feedback

Brief Description

Users (students) can provide feedback to the counterparty after the transaction is completed to build trust in the platform.

Initial Step-By-Step Description

When the use case is activated, the user has logged in and completed the transaction.

1. The user selects ‘Give Feedback’ on the Transaction History page.

2. The system displays a feedback form that allows the user to rate and enter specific comments.

3. The user completes the rating and text feedback.

4. the user submits the feedback.

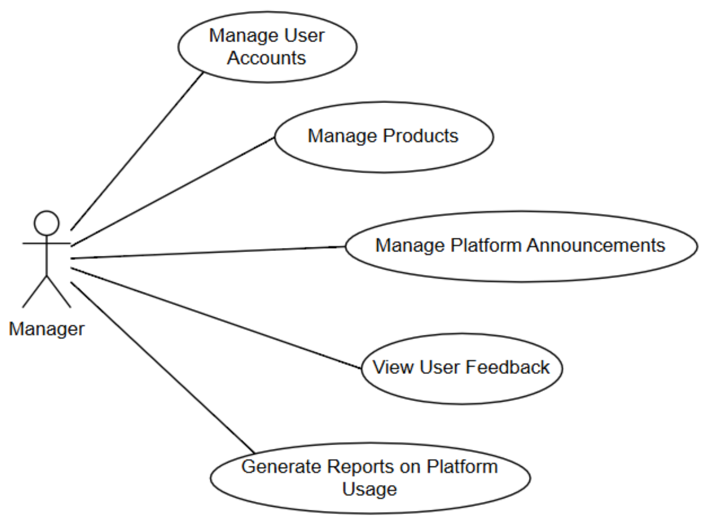
5. The system saves the feedback to the database and displays it in the counterparty's rating history.

Alternative Paths

If the user has already submitted feedback on the transaction, the system displays a 'Feedback Submitted' message and prohibits duplicate submissions.

3.2.2 Manager Use Cases

Diagram:



Use Case: Manage User Accounts

Brief Description

Administrators can view and manage user accounts on the platform, including blocking or unblocking unauthorized accounts.

Initial Step-By-Step Description

Before this use case is launched, the administrator has logged in and entered the administrator console.

1. The administrator selects Manage User Accounts.

2. The system displays a list of all user accounts.

3. The administrator selects a specific account to view.

4. The system displays the user's details and action options (e.g., blocking, unblocking).

5. The administrator performs actions as required (e.g., blocking an account).

Alternative Paths

If the account is blocked, the system displays the blocked status and allows unblocking.

If the account selected by the administrator does not exist, an error message is displayed.

Use Case: Manage Products

Brief Description

Administrators can manage the list of items on the platform, including checking, editing, or deleting inappropriate items to ensure compliance and keep the platform tidy.

Initial Step-By-Step Description

When the use case starts, the administrator logs in and enters the administrator console and selects 'Manage Items'.

1. The administrator selects a list of items (e.g. Sale, Barter and Rental) to be managed.

2. The system displays a list of all published items, including basic information such as the item name, category, publisher, and publish date.

4. The administrator selects a specific item to view the details.

5 The system displays detailed information about the item and options for managing the item (such as editing, deleting 6. or flagging it as a violation).

7. If an item is determined to be inappropriate, the administrator can choose to delete it or issue a warning to the publisher.

8. The system records the administrator's action and notifies the publisher of the item, if necessary.

Alternative Paths

If the item has been processed by another administrator, the system shows the processed status and records the latest status of the item.

The administrator can cancel the current action, the system will return the item list without any change.

Use Case: Manage Platform Announcements

Brief Description

Administrators can post platform announcements to notify users of important information or system updates.

Initial Step-By-Step Description

Before this use case is started, the administrator has logged in and accessed the administrator console.

1. The administrator selects Manage Platform Announcements.

2. The system displays a list of existing announcements.

3. The administrator chooses to add a new announcement or edit an existing announcement.

4. The system displays the announcement editor.

5. The administrator enters the contents of the announcement and selects Publish.

6. The system displays the announcement in the user notification area.

Alternative Paths

The administrator can choose to edit or delete the existing announcement, and the system updates the status of the announcement.

Use Case: View User Feedback

Brief Description

Administrators can view all user feedback, monitor evaluations, and ensure that feedback complies with the platform's specifications.

Initial Step-By-Step Description

When the use case starts, the administrator logs in and enters the administrator console.

1. The administrator selects View User Feedback.

2. The system displays a list of all user feedback, including ratings, comments, submitters, and target audience.

3. The administrator selects specific feedback to view the details.

4. The system displays the complete feedback with associated options (e.g., mark as inappropriate, contact the rating provider).

5. The administrator decides whether or not to process the feedback as needed.

Alternative Paths

If the feedback has been processed by another administrator, this status is displayed.

Postcondition

The administrator completes viewing the feedback and handles inappropriate feedback, such as flagging or deleting it, as needed.

Use Case: Generate Reports on Platform Usage

Brief Description

Administrators can generate reports on platform usage, including metrics such as user activity, number of transactions, and number of content reports.

Initial Step-By-Step Description

Before this use case is launched, the administrator logs in and enters the administrator console.

1. The administrator selects Generate Platform Report.

2. The system displays the available report types (e.g., User Activity, Transaction Volume, Number of Reports).

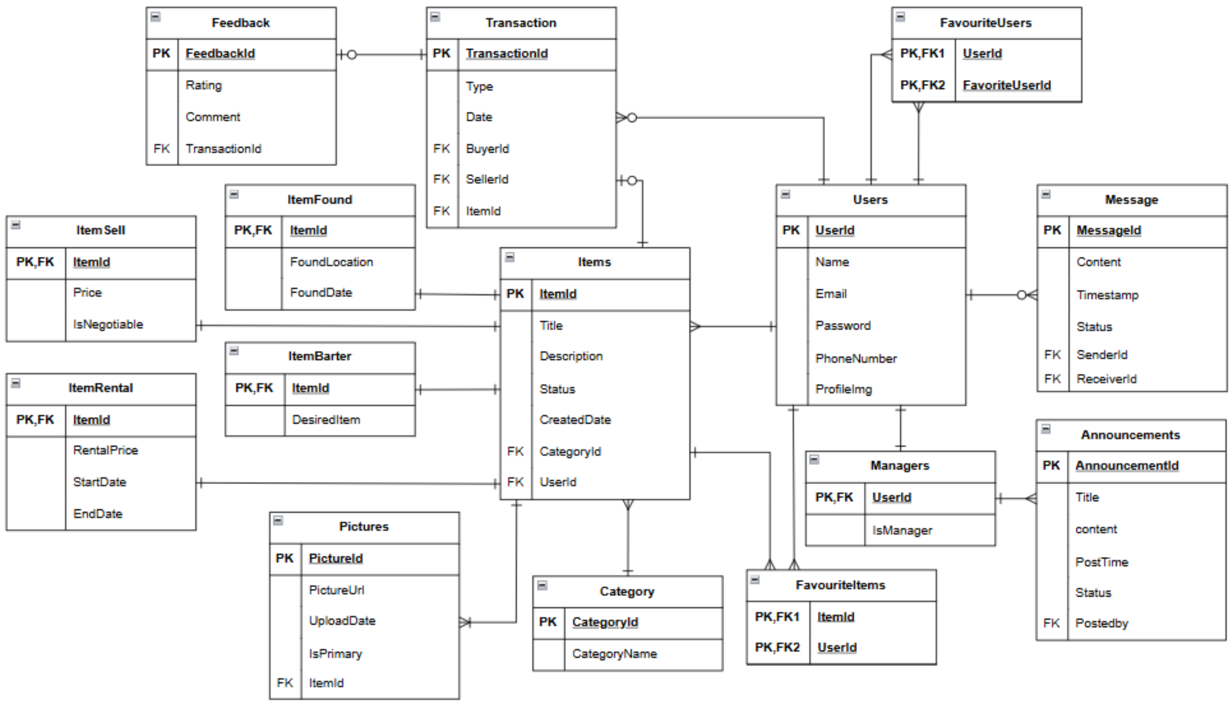
3. The administrator selects the desired report type and clicks Generate.

4. The system aggregates the data and generates the report.

5. The administrator can choose to download or view the report content.

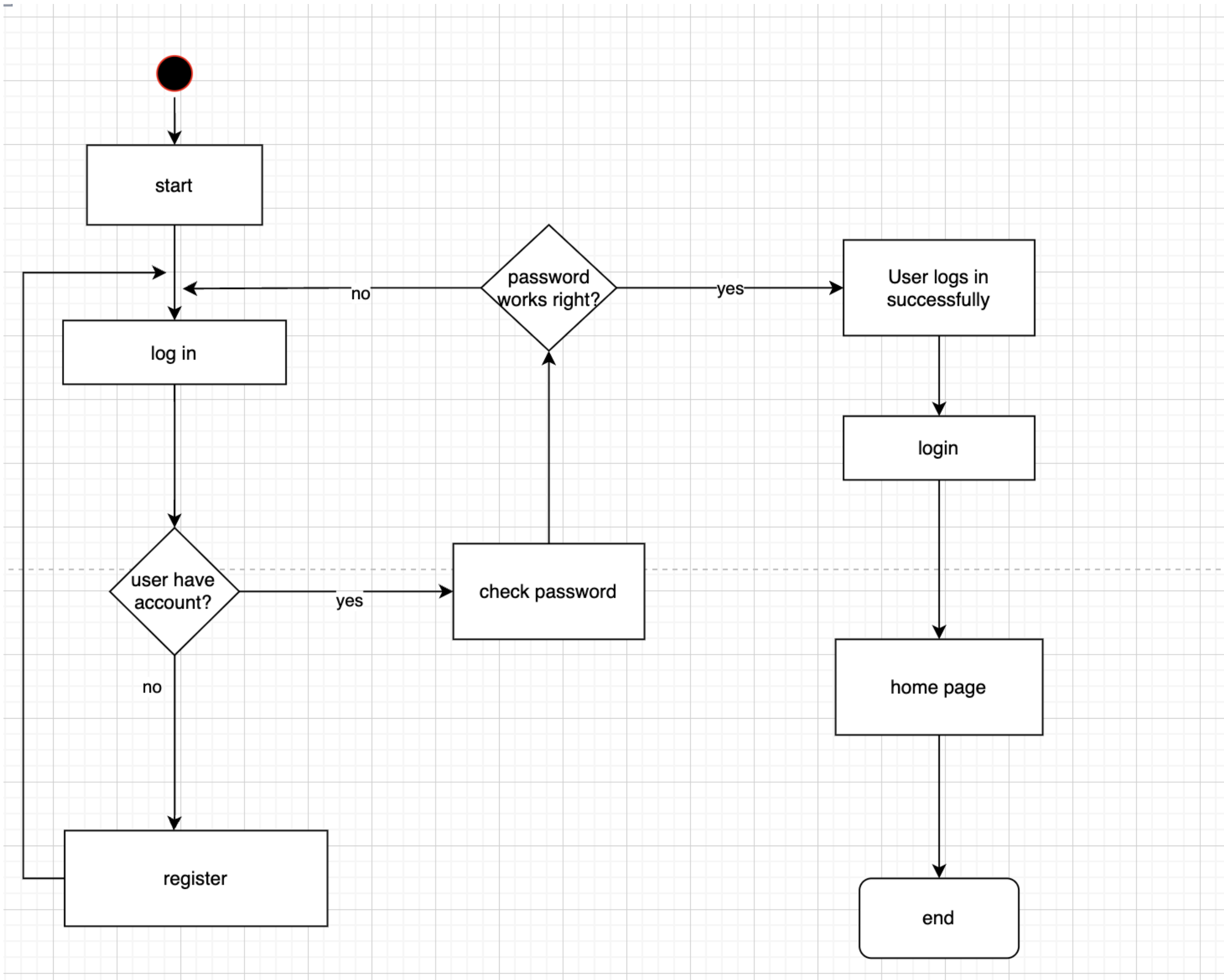
3.3 Data Modelling and Analysis

3.3.1 Normalized Data Model Diagram

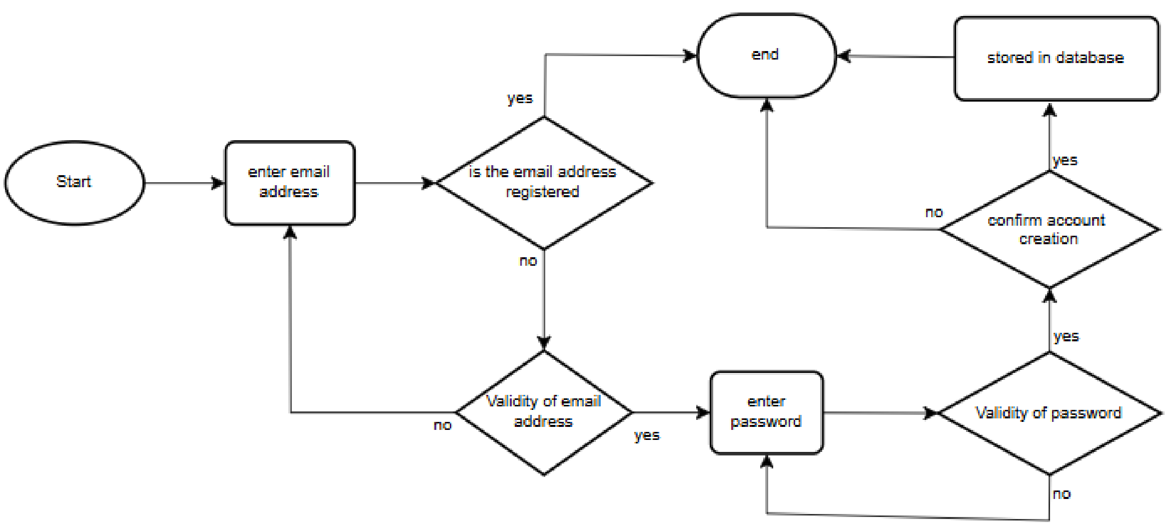


3.3.2 Activity Diagrams

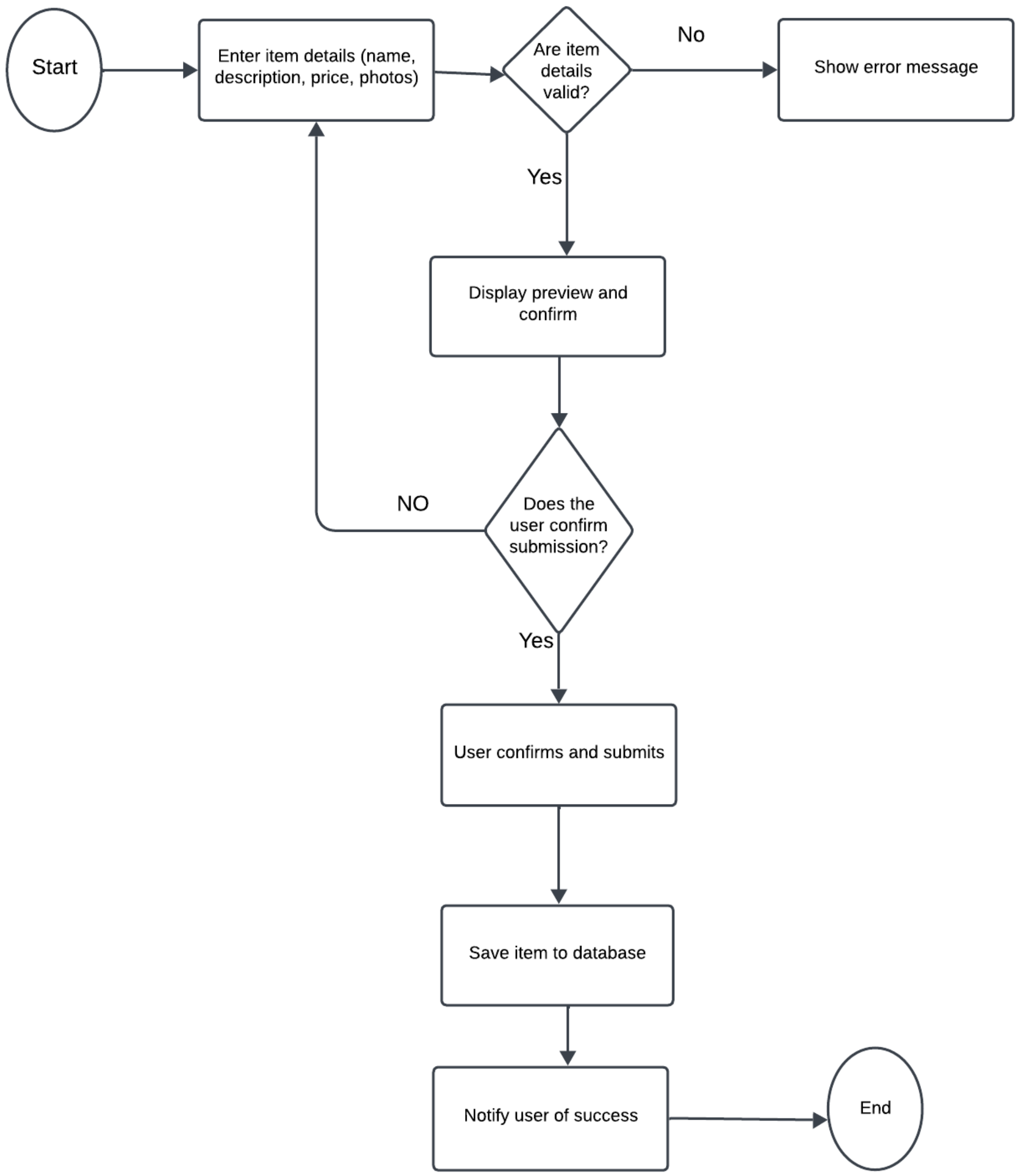
1. **Login Process Activity Diagram**



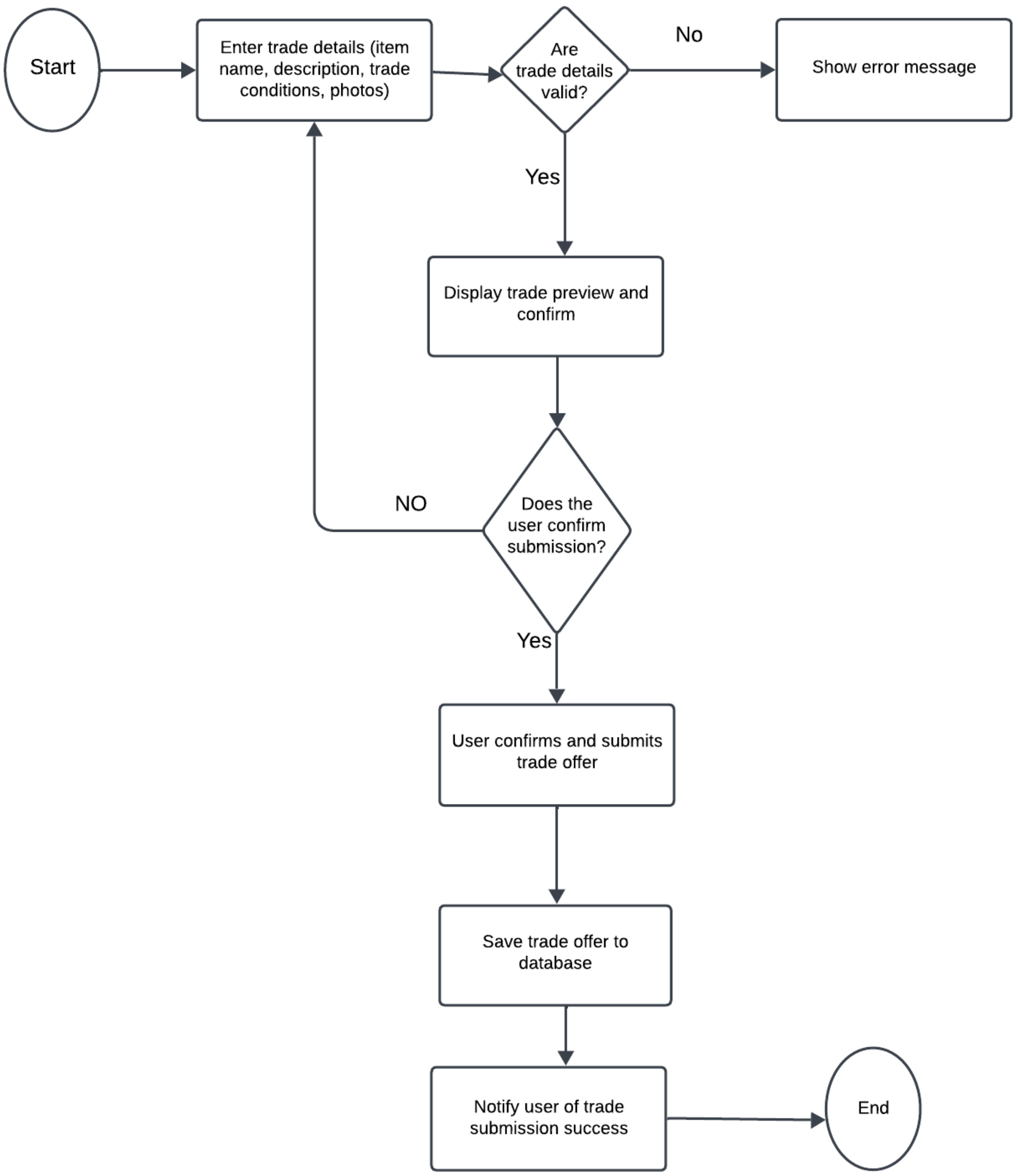
2. Creating Account Activity Diagram



3. Posting Items Activity Diagram

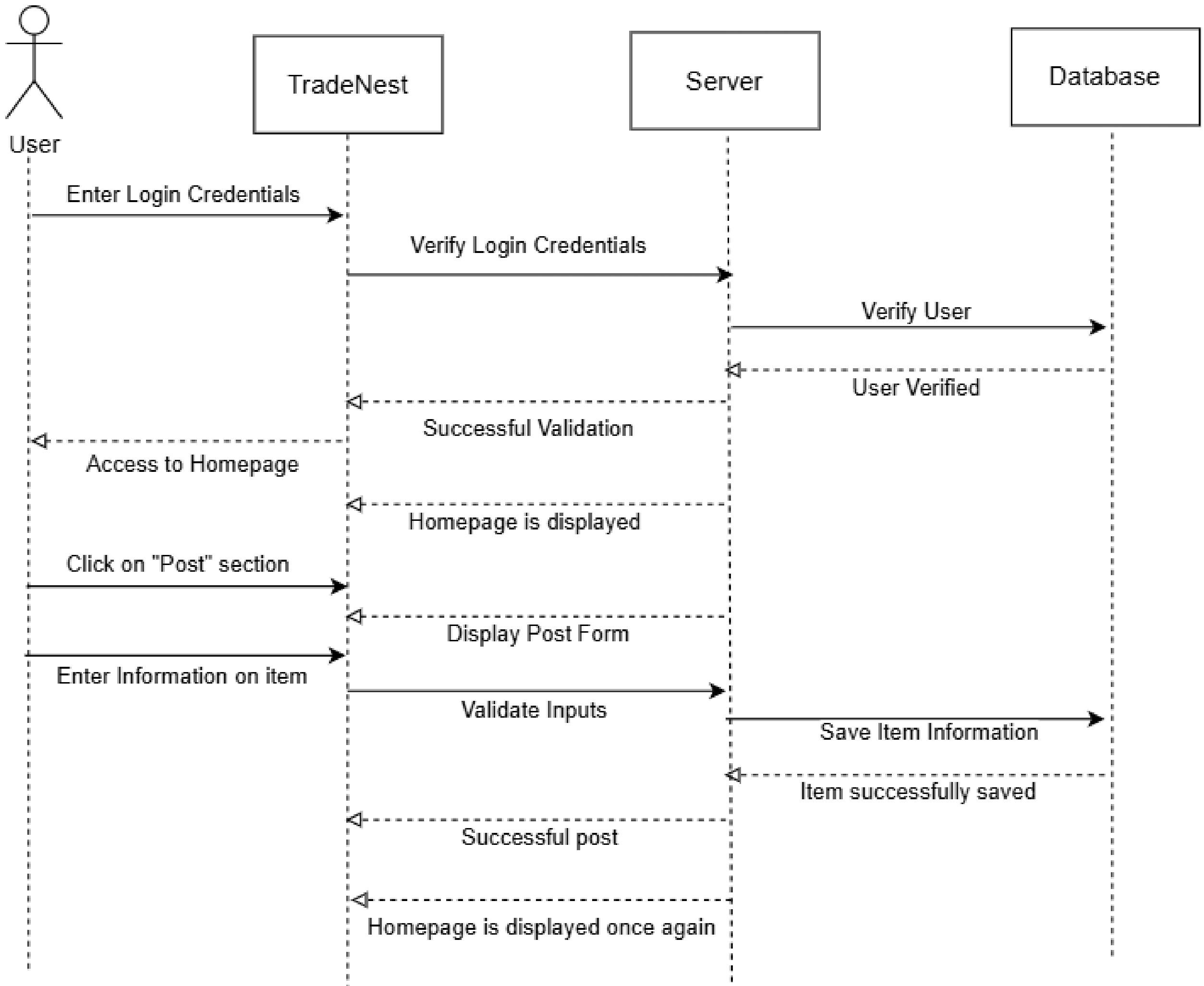


4. Traded Items Activity Diagram

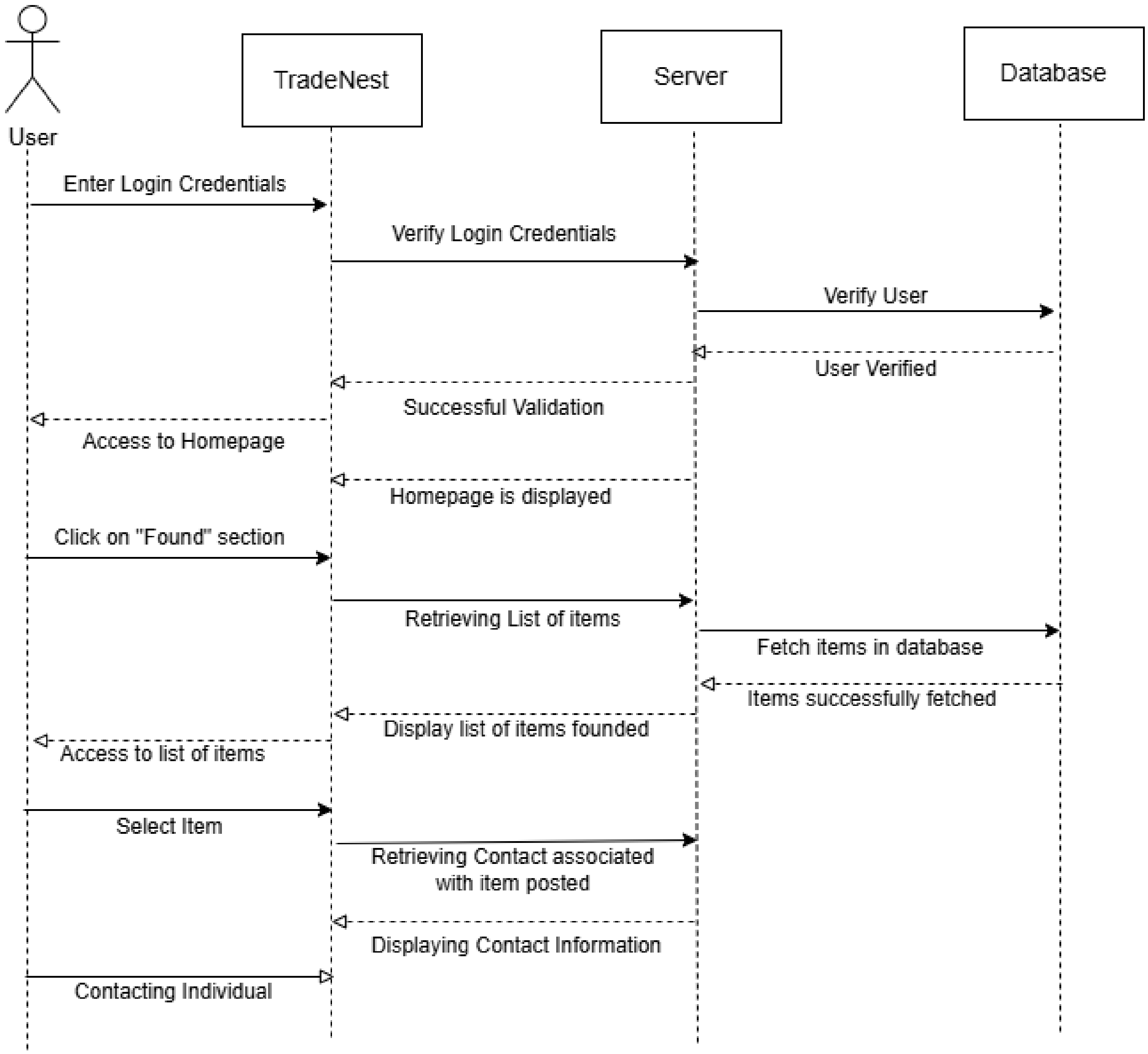


3.3.3 Sequence Diagrams

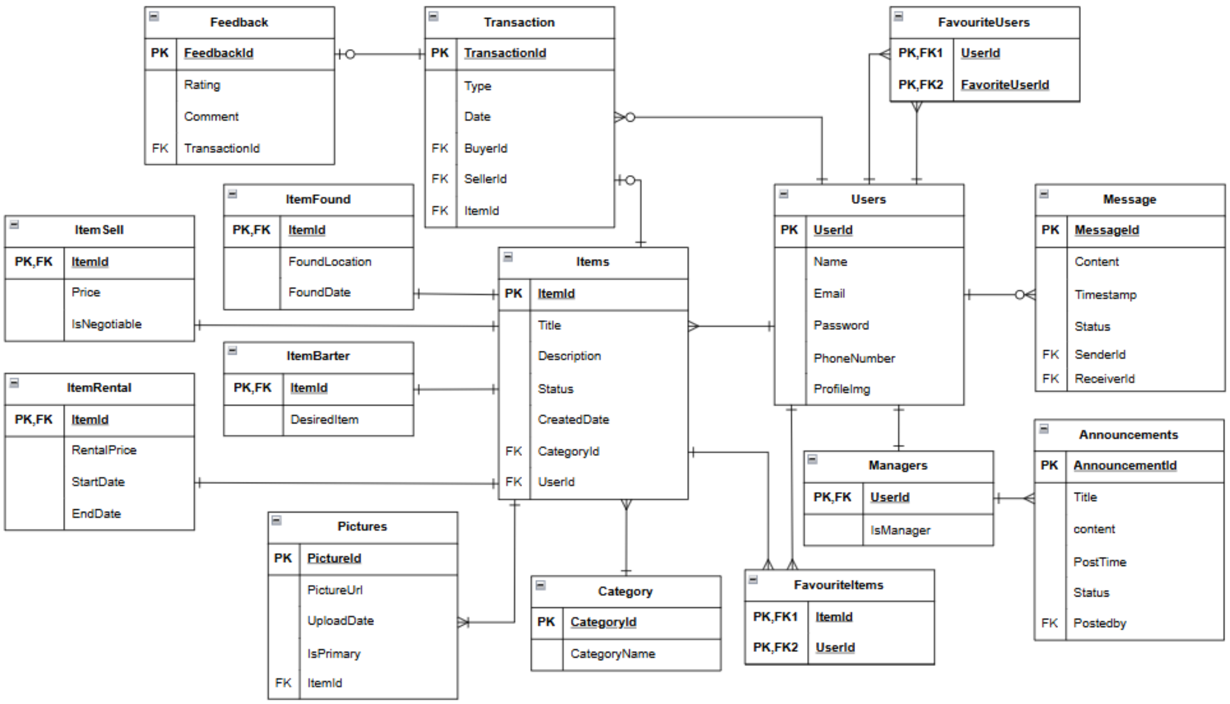
Displayed here is the sequence diagram of a user uploading a new post on an item found:



Displayed here is the sequence diagram of a user reclaiming their lost item:

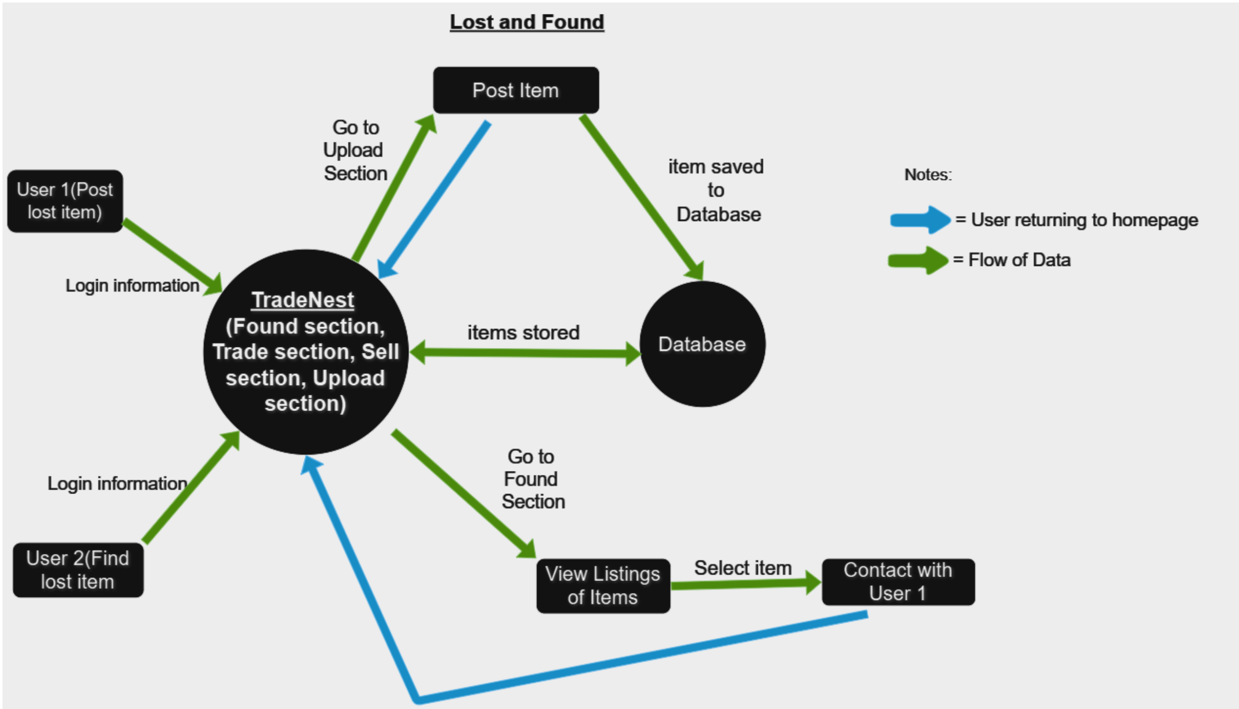

User
TradeNest
Server
Database
Enter Login Credentials
Verify Login Credentials
Verify User
User Verified
Successful Validation
Access to Homepage
Homepage is displayed
Click on "Found" section
Retrieving List of items
Fetch items in database
Items successfully fetched
Display list of items founded
Access to list of items
Select Item
Retrieving Contact associated
with item posted
Displaying Contact Information
Contacting Individual

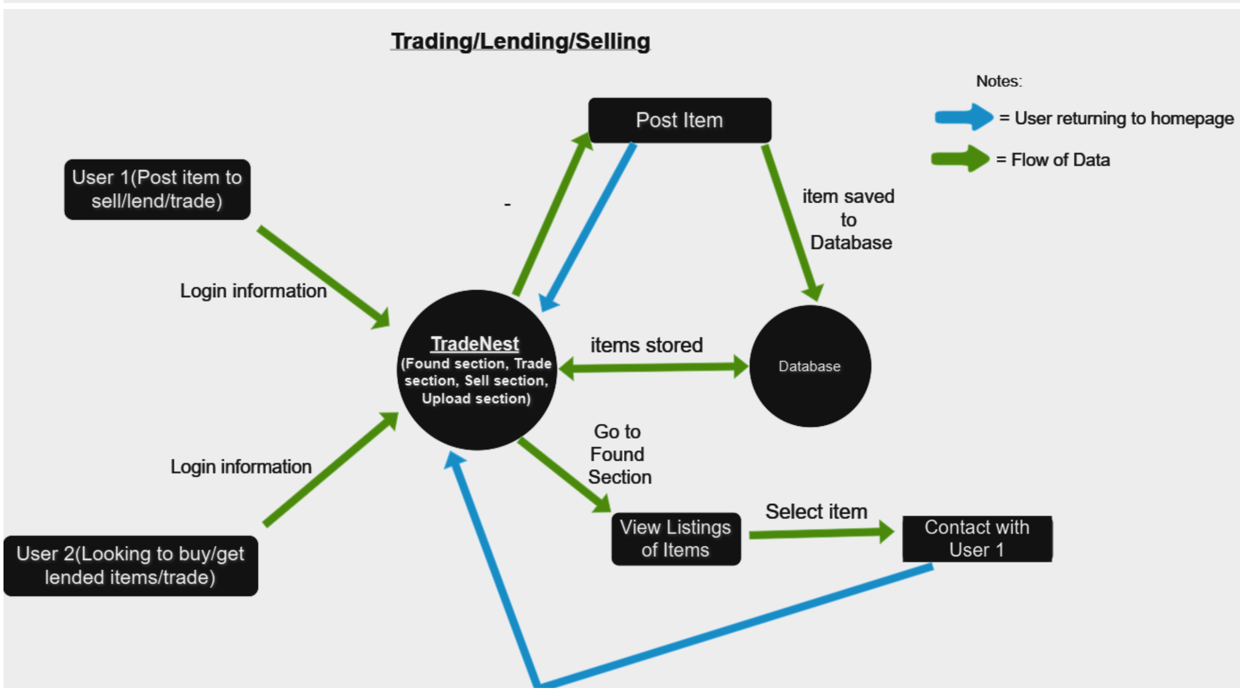
3.3.4 UML Class Diagram



3.4 Process Modelling

Both display the same functionality but with different tasks, depending on the user.





4. Non-Functional Requirements

1. Performance:

- The system should handle up to 500 concurrent users without degradation in response time.

- Page load times should not exceed 2 seconds under normal conditions.

- Search queries should return results within 1 second.

2. Reliability:

- The system should have an uptime of at least 99.5% over any given month.

- Users should be able to access their accounts and data without error at all times, barring scheduled maintenance.

3. Availability:

- TradeNest must be available to users 24/7, with maintenance periods scheduled during low-traffic hours.

4. Security:

- All user data, including messages and transactions, should be encrypted both at rest and in transit.

- Only users with a George Brown College email address should be able to register and access the platform.

- The system must perform regular vulnerability scans and implement multi-factor authentication (MFA) for additional security.

5. Maintainability:

- System updates or patches should be applied with minimal downtime, ideally during off-peak hours.

- Code should be modular to allow for easy updates or modifications to specific components without impacting the entire system.

6. Portability:

- TradeNest should be accessible on desktop and mobile devices, with optimized views for both platforms.

- The system must support at least two major web browsers (e.g., Chrome and Safari) on both iOS and Android.

5. Logical Database Requirements

1, User Data Entity

Represents a registered user on the platform, restricted to users with institutional emails.

Data Item	Type	Description	Comments
User ID	Integer	Unique identifier for each user	Primary Key
Name	Text	Full name of the user	
Email	Text	Institutional email address	Must end with @georgebrown.ca
Password	Text	Encrypted password	Security: stored as hash
Profile Picture	Text	URL of the user's profile picture	Optional
Bio	Text	Short user bio or description	Optional
Contact Info	Text	Contact details (phone)	Optional
Following Users	Array	List of User IDs the user is following	References User ID in User

2. Item Entity

Represents items posted by users, which can be marked as found, for sale, for trade, or for lending.

Data Item	Type	Description	Comments
Item ID	Integer	Unique identifier for each item	Primary Key
User ID	Integer	ID of the user who posted the item	Foreign Key, references User ID in User
Title	Text	Name/title of the item	
Description	Text	Detailed description of the item	
Category	Text	Item category (e.g., Electronics, Clothing)	
Condition	Text	Condition of the item (e.g., New, Used)	
Availability Status	Text	Status of the item (e.g., Found, Available, Taken)	
Images	Array	URLs of images related to the item	Optional
Created At	Date	Date and time item was posted	
Location	Text	Location where the item is or was found	Optional

3. Message Entity

Represents messages exchanged between users within the app.

Data Item	Type	Description	Comments
Message ID	Integer	Unique identifier for each message	Primary Key
Sender ID	Integer	User ID of the sender	Foreign Key, references User ID in User
Receiver ID	Integer	User ID of the receiver	Foreign Key, references User ID in User
Content	Text	Text content of the message	
Sent At	Date	Date and time message was sent	
Read Status	Boolean	Indicates if the message has been read	Default: False

4. Feedback Entity

Represents user-provided feedback after an in-person exchange, lending, or sale to build trust within the platform.

Data Item	Type	Description	Comments
Feedback ID	Integer	Unique identifier for each feedback entry	Primary Key
Item ID	Integer	ID of the item related to the feedback	Foreign Key, references Item ID in Item
From User ID	Integer	ID of the user providing the feedback	Foreign Key, references User ID in User
To User ID	Integer	ID of the user receiving the feedback	Foreign Key, references User ID in User
Rating	Integer	Rating score provided (1-5)	
Comment	Text	Feedback comments	Optional
Created At	Date	Date and time feedback was created	

6. Approval

Project Role	Name	Signature	Date
Lead Developer	Kaman Wong	K.W.	11/06/2024
UX Designer	Naomi Teklu	N.T.	11/06/2024
Project Manager	Jason Opoku	J.O.	11/06/2024
Security	Shalom Aideyan	S.A.	11/06/2024